

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления  
Кафедра Интеллектуальных информационных технологий

*К защите допустить:*

Заведующий кафедрой

\_\_\_\_\_ Д. В. Шункевич

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

по дисциплине «Проектирование программ в интеллектуальных системах»:

**Нахождение компонент связности в  
неориентированном графе**

КРЗ 1–40 03 01 01 102 ПЗ

Студент:

Е. С. Стефаненко

Руководитель:

Д. В. Шункевич

Минск 2020

## Содержание

Перечень условных обозначений . . . . .	5
Введение . . . . .	6
1 Теоретико-графовая задача . . . . .	7
1.1 Список основных понятий предметной области . . . . .	7
1.2 Разработка алгоритма . . . . .	7
1.3 Тестовые примеры . . . . .	7
1.4 Пример работы алгоритма в семантической памяти . . . . .	12
2 Личный вклад в развитие ИС по искусству . . . . .	21
2.1 Список формализованных понятий . . . . .	21
2.2 Примеры реализации формализованных понятий . . . . .	22
Заключение . . . . .	26
Список использованных источников . . . . .	27

## **Перечень условных обозначений**

В курсовой работе используются следующие условные обозначения:

ЯП – язык программирования;

ИС – интеллектуальная система;

SC – Semantic Code;

SCg – Semantic Code graphical;

SCp – Semantic Code programming.

## Введение

Целями курсовой работы по предмету «Проектирование программ в интеллектуальных системах» в данном семестре является анализ и решение поставленной задачи: нахождение компонент связности в неориентированном графе. Решение требуется реализовать в двух видах:

1. В виде программы с использованием библиотеки, поддерживающей SC-память, на классическом ЯП (C++);
2. В виде программы на ЯП SCp: языке, ориентированном на обработку семантических сетей.

Для достижения этих целей были поставлены следующие задачи: изучение теоретического материала о графах; создание алгоритма решения поставленной теоретико-графовой задачи; изучение необходимых языков программирования, их структуры, семантики, представление информации в памяти.

# 1 Теоретико-графовая задача

## 1.1 Список основных понятий предметной области

1. Граф (абсолютное понятие) – совокупность непустого множества вершин и наборов пар вершин (связей между вершинами).
2. Неориентированный граф (абсолютное понятие) – граф, в котором все связки-ребра.
3. Связный граф (абсолютное понятие) – граф, содержащий только одну компоненту связности.
4. Компонента связности - максимальный связный подграф графа.

## 1.2 Разработка алгоритма

Для решения задачи по нахождению компонент связности в неориентированном графе потребовалось изучить необходимые сведения из теории графов о графе, неориентированном графе и его свойствах, связном графе, компонентах связности.

## 1.3 Тестовые примеры

Во всех тестах графы будут приведены в сокращенной форме со скрытыми ролями элементов графа.

### 1.3.1 Тест 1

**Вход:**

Найти компоненты связности графа. Входные данные представлены на рис. 1.1.

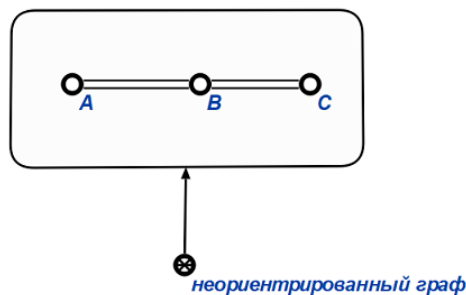


Рисунок 1.1 – Вход теста 1

**Выход:**

Выходные данные представлены на рис. 1.2

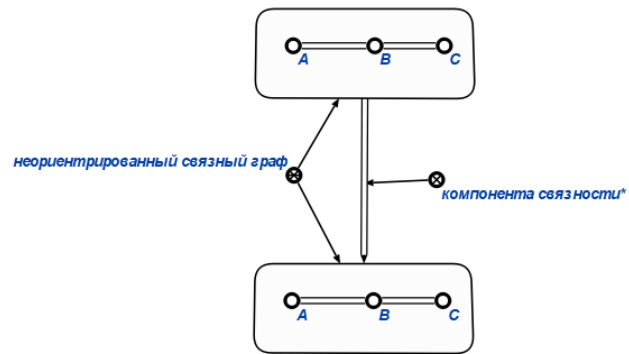


Рисунок 1.2 – Выход теста 1

### 1.3.2 Тест 2

**Вход:**

Найти компоненты связности графа. Входные данные представлены на рис. 1.3.

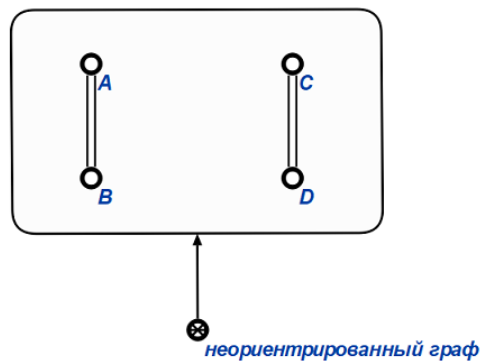


Рисунок 1.3 – Вход теста 2

**Выход:**

Выходные данные представлены на рис. 1.4.

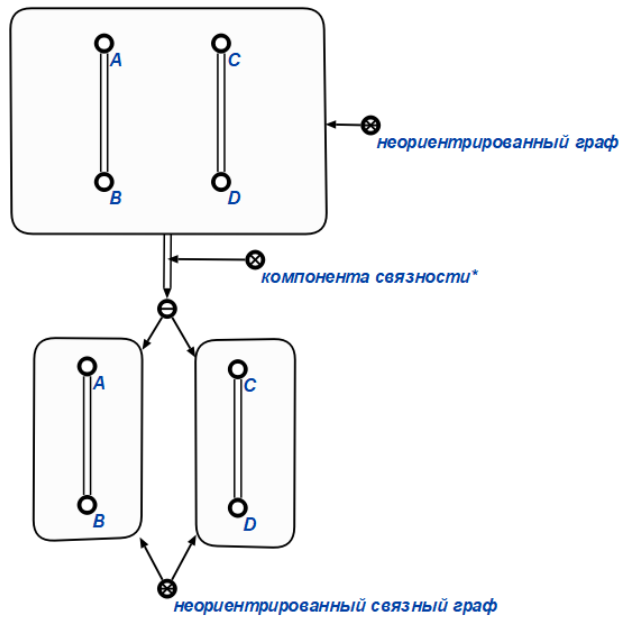


Рисунок 1.4 – Выход теста 2

### 1.3.3 Тест 3

#### Вход:

Найти компоненты связности графа. Входные данные представлены на рис. 1.5.

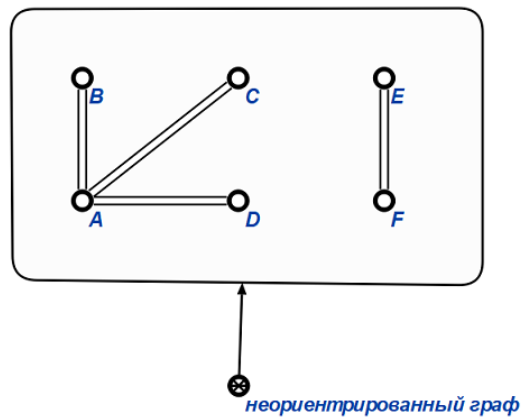


Рисунок 1.5 – Вход теста 3

#### Выход:

Выходные данные представлены на рис. 1.6.

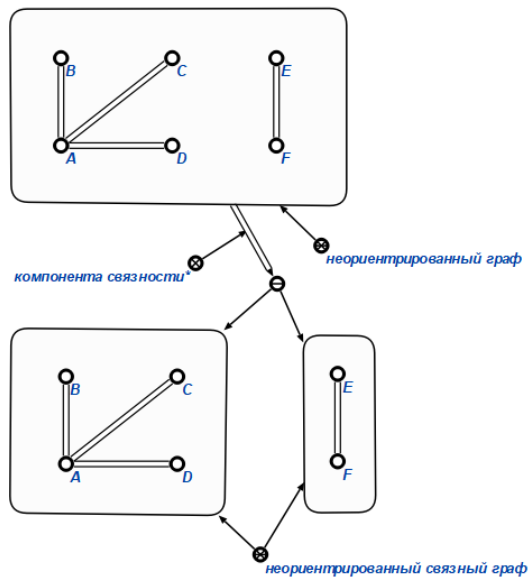


Рисунок 1.6 – Выход теста 3

#### 1.3.4 Тест 4

##### Вход:

Найти компоненты связности графа. Входные данные представлены на рис. 1.7.

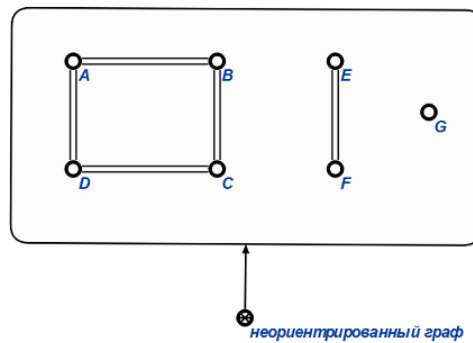


Рисунок 1.7 – Вход теста 4

##### Выход:

Выходные данные представлены на рис. 1.8.



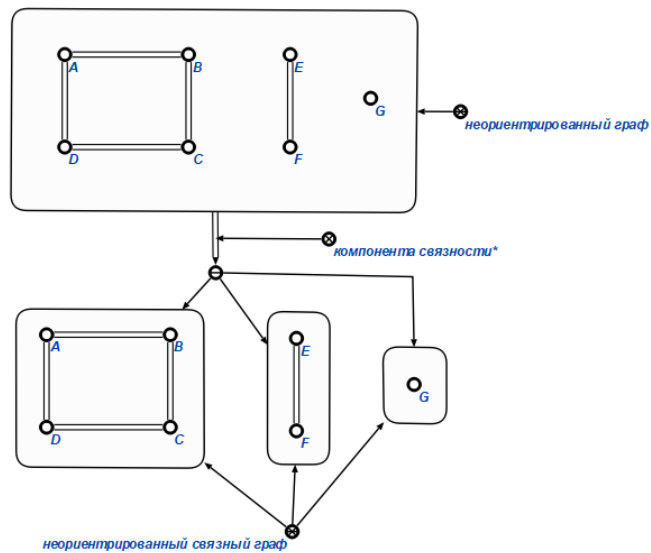


Рисунок 1.8 – Выход теста 4

### 1.3.5 Тест 5

**Вход:**

Найти компоненты связности графа. Входные данные представлены на рис. 1.9.

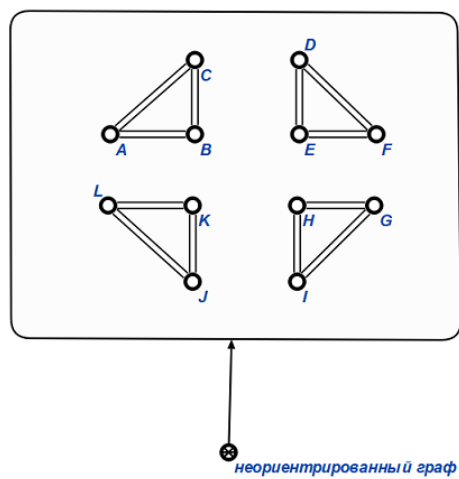


Рисунок 1.9 – Вход теста 5

**Выход:**

Выходные данные представлены на рис. 1.10.

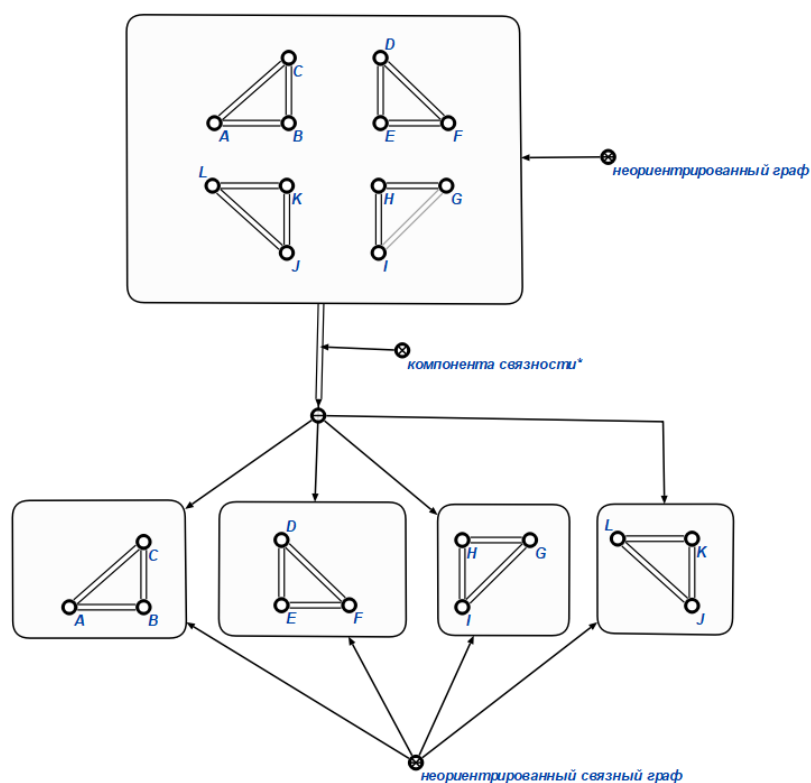


Рисунок 1.10 – Выход теста 5

## 1.4 Пример работы алгоритма в семантической памяти

### 1.4.1 Описание алгоритма

1. Добавить все вершины графа во множество нерассмотренных вершин.
2. Начало формирования нового подграфа, который является компонентой связности:
  - 2.1 Создать новое множество вершин подграфа.
  - 2.2 Добавить в множество вершин подграфа одну из вершин из множества нерассмотренных вершин. Считать эту вершину рассматриваемой.
  - 2.3 Удалить рассматриваемую вершину из множества нерассмотренных вершин.
3. Добавление элемента в множество вершин подграфа:
  - 3.1 Новым элементом множества вершин подграфа является вершина из множества нерассмотренных вершин, смежная рассматриваемой вершине.
  - 3.2 Если вершина попала в множество вершин подграфа, то удалить ее из множества нерассмотренных вершин.
  - 3.3 Далее считать данную вершину рассматриваемой.

4. Если для вершин, принадлежащих множеству вершин подграфа, нет смежных вершин из множества нерассмотренных вершин, то завершить формирование множества вершин подграфа, иначе перейти к пункту 3.

5. Завершение формирования подграфов.

5.1 Если множество нерассмотренных вершин не пустое, то перейти к пункту 2.

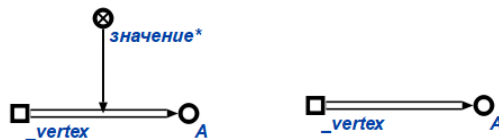
5.3 Сформировать из множеств вершин подграфа и соединяющих их ребер новые неориентированные связные графы.

5.4 Завершить алгоритм.

### 1.4.2 Пример выполнения алгоритма в сс-памяти

#### Соглашения по демонстрации

Для наглядности примеры формализации переменных и их значений будут представлены в кратком виде (опуская отношение значение\*).



Для решения задачи необходимы следующие переменные:

1. `_graph`, содержащая узел исходного неориентированного графа.
2. `_subgraph_i` (где  $i=1,2,3\dots$ ), содержащая узел неориентированного связного графа, являющегося подграфом исходного графа.
3. Множество нерассмотренных вершин `_not_explored_vertices`.
4. Множество вершин подграфа `_vertices_of_subgraf_i` (где  $i=1,2,3\dots$ ).
5. `_current_vertex`, содержащая рассматриваемую вершину.

#### Демонстрация алгоритма

Исходные данные - некий неориентированный граф (рис. 1.11).

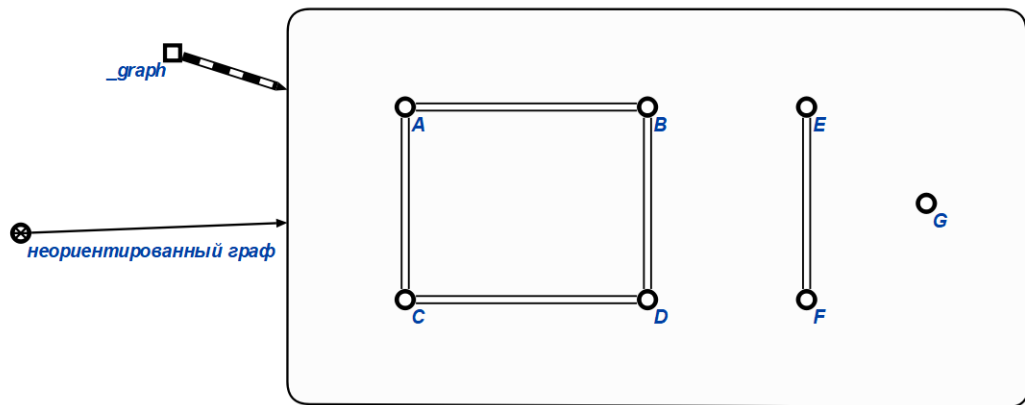


Рисунок 1.11 – Исходные данные

### Шаг 1

Добавление всех вершин графа во множество `_not_explored_vertices` (рис. 1.12).

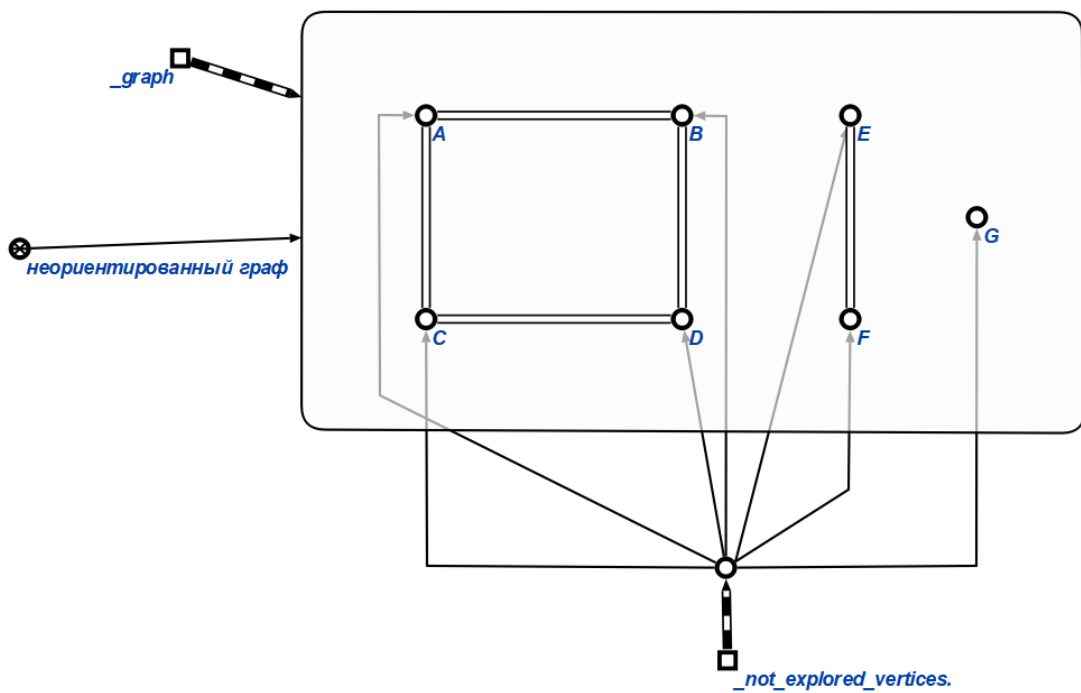


Рисунок 1.12 – Шаг 1

## Шаг 2

Начало формирования нового множества `_vertices_of_subgraph_1` (рис. 1.13).

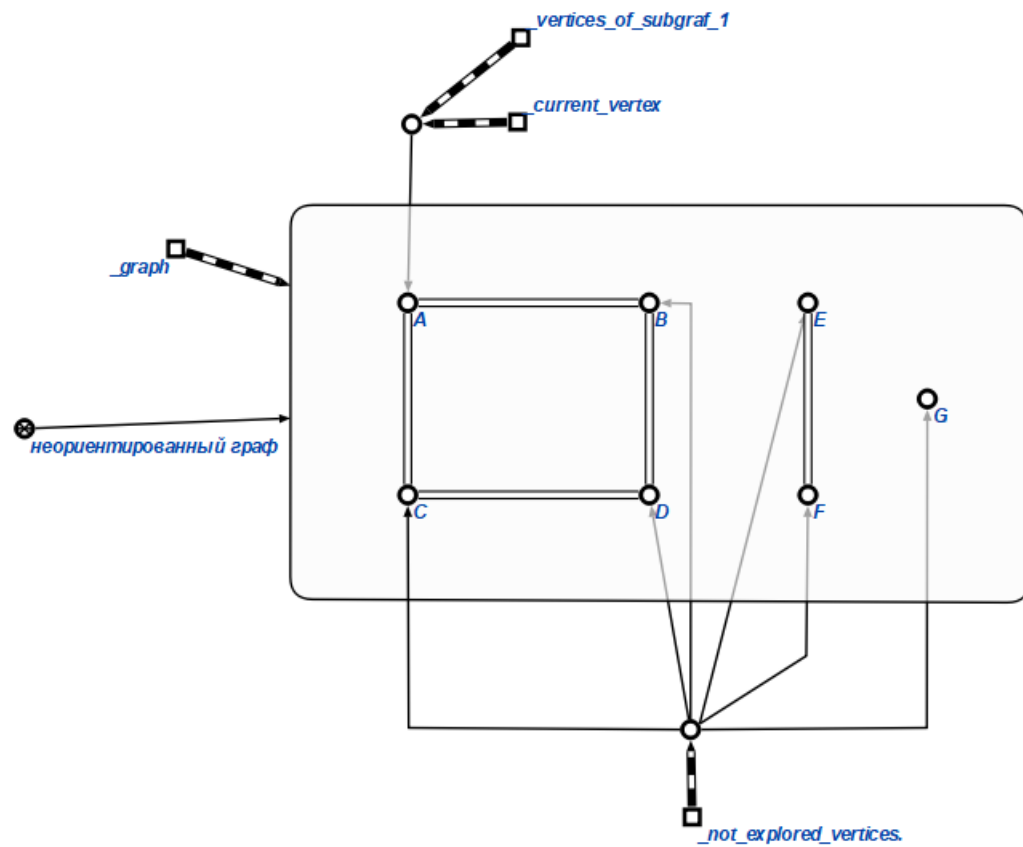


Рисунок 1.13 – Шаг 2

## Шаги 3,4,5

Добавление элементов во множество `_vertices_of_subgraph_1` (рисунки 1.14, 1.15, 1.16).

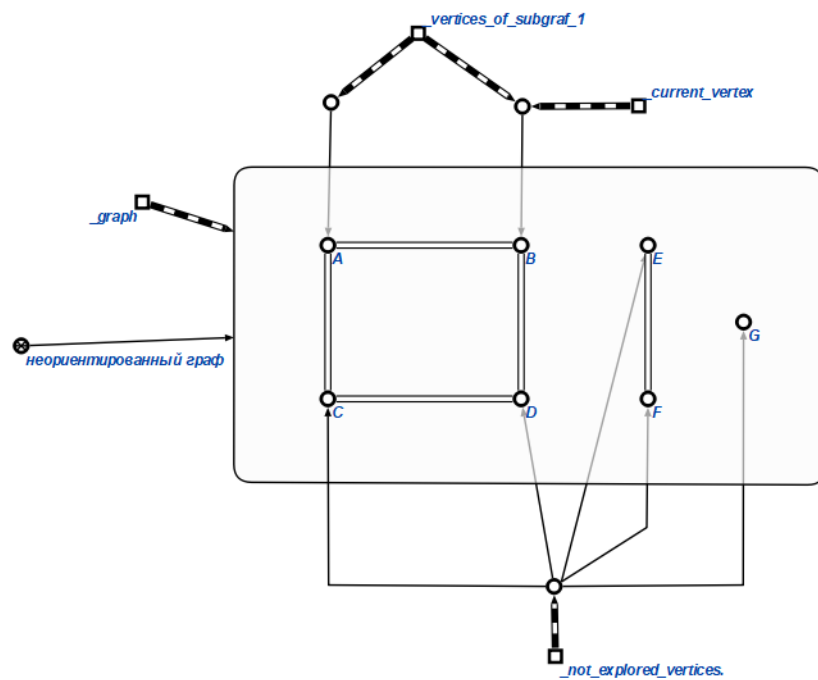


Рисунок 1.14 – Шаг 3

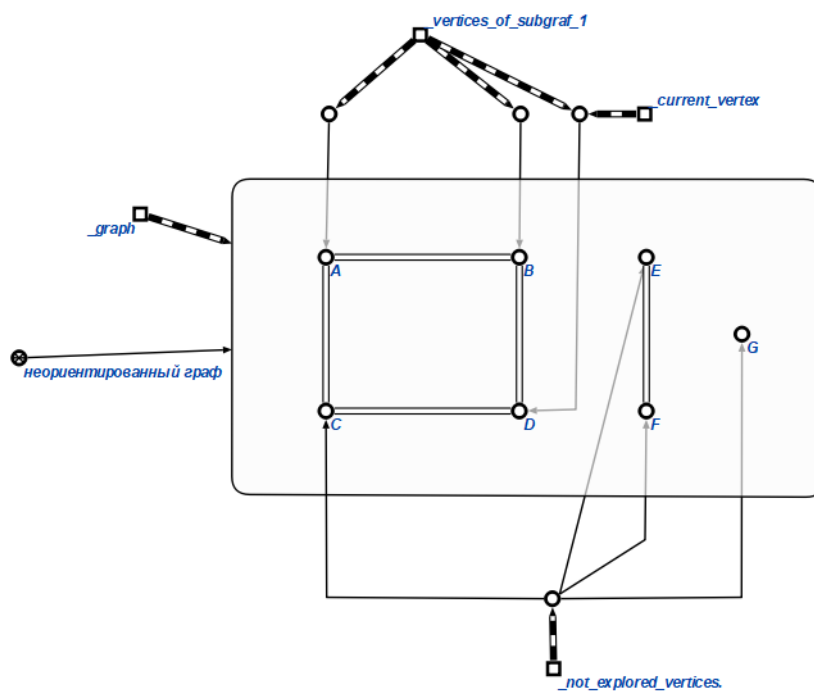


Рисунок 1.15 – Шаг 4

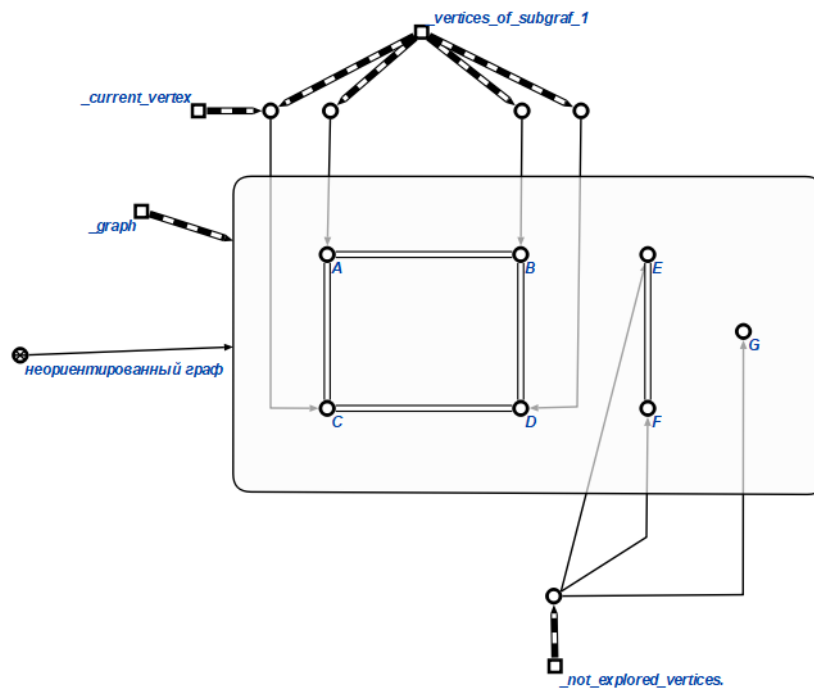


Рисунок 1.16 – Шаг 5

### Шаг 6

Начало формирования нового множества `_vertices_of_subgraph_2` (рис. 1.17).

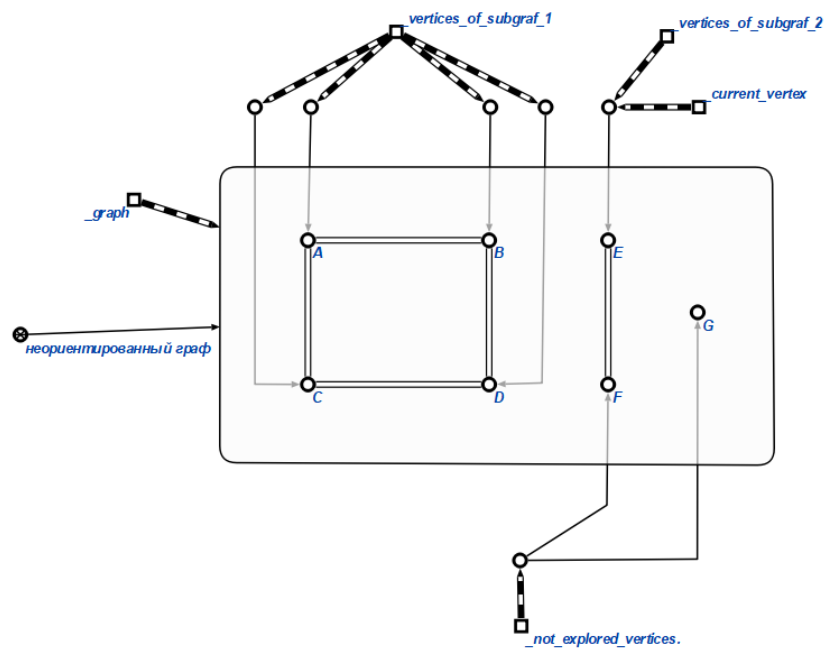


Рисунок 1.17 – Шаг 6

## Шаг 7

Добавление элемента во множество `_vertices_of_subgraph_2` (рис. 1.18).

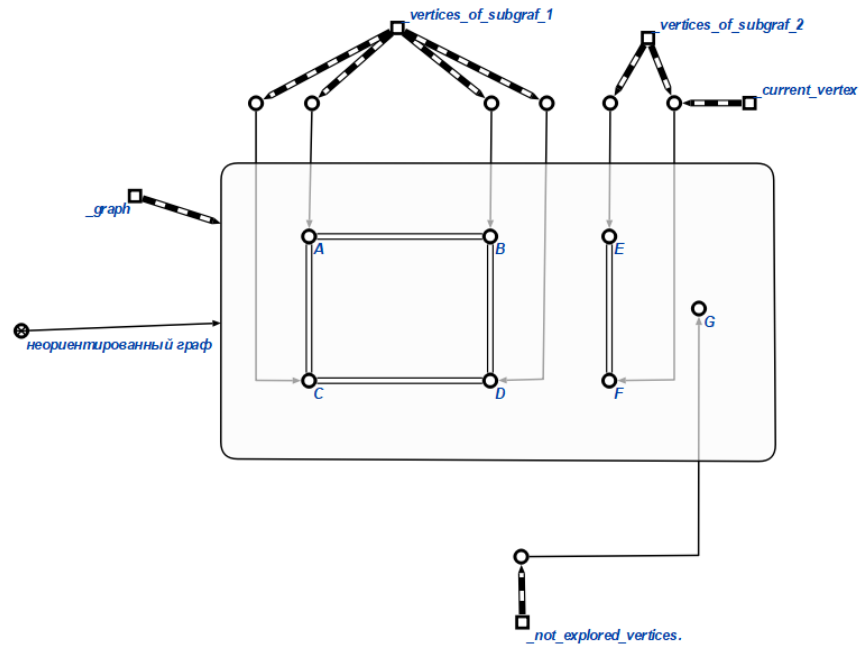


Рисунок 1.18 – Шаг 7

## Шаг 8

Начало формирования нового множества `_vertices_of_subgraph_3` (рис. 1.19).

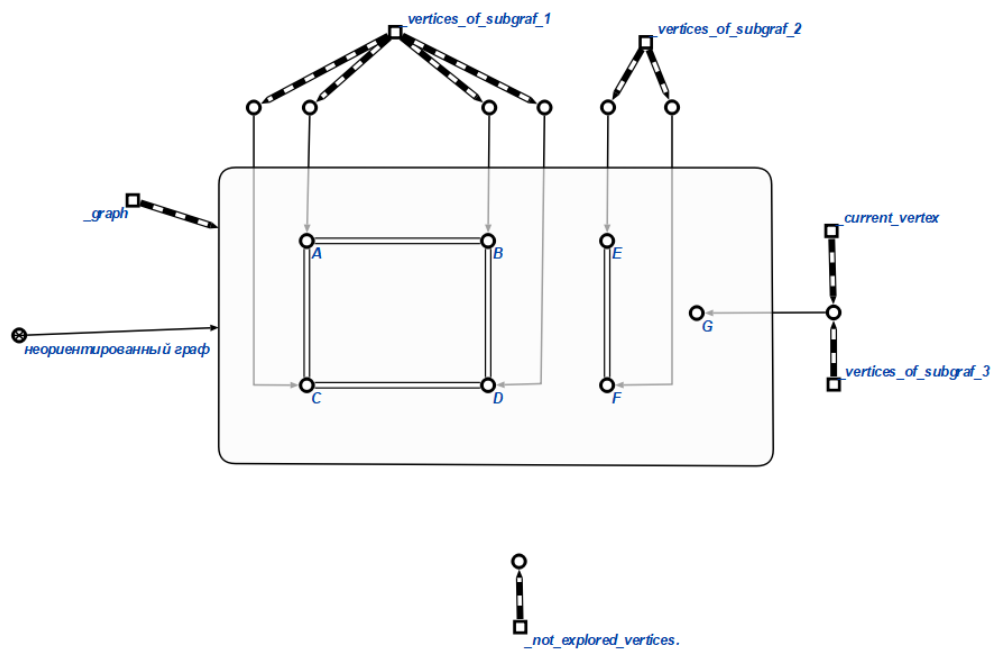


Рисунок 1.19 – Шаг 8



## Шаги 9, 10

Завершение формирования подграфов (рис. 1.20). Завершение алгоритма (рис. 1.21).

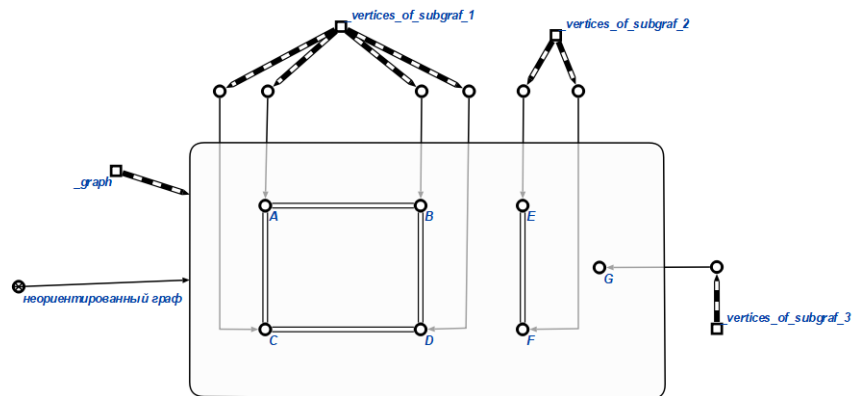


Рисунок 1.20 – Шаг 9

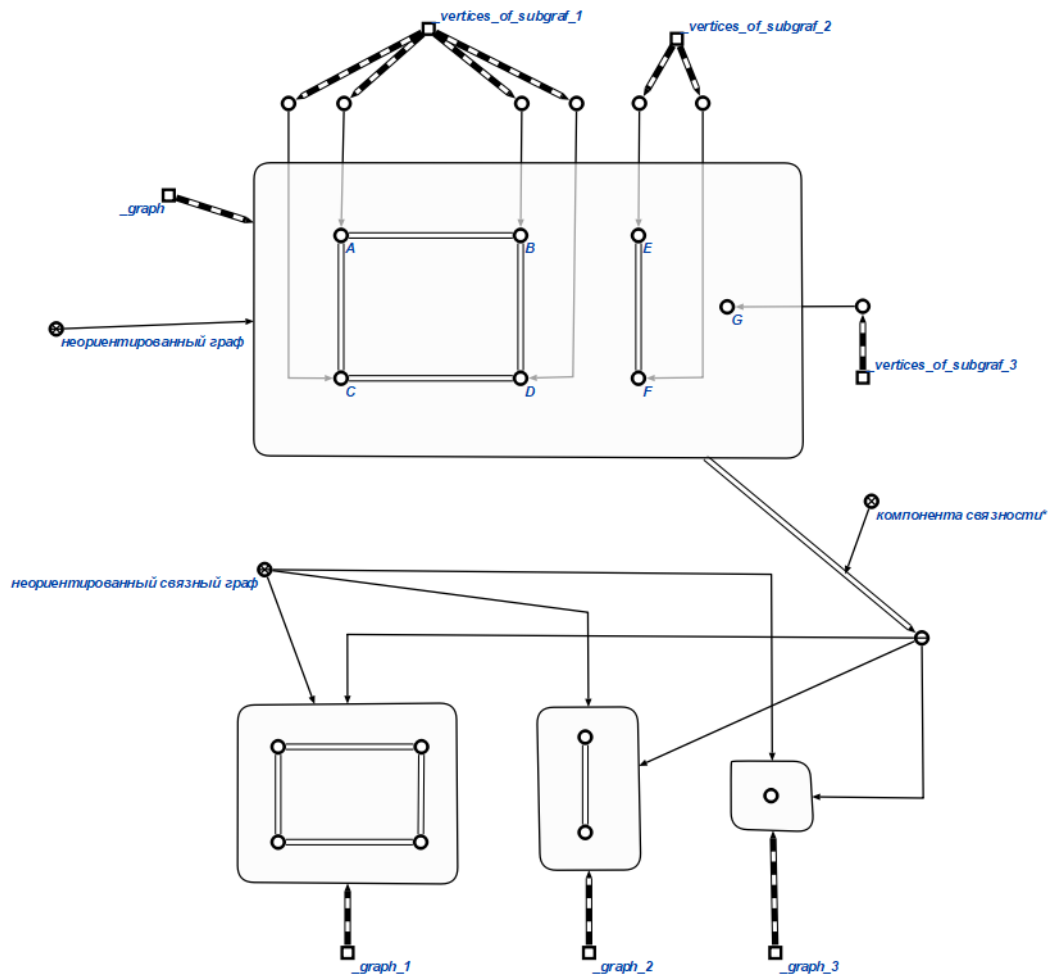


Рисунок 1.21 – Шаг 10

## Шаг 11

Вывод окончательного результата (рис. 1.22).

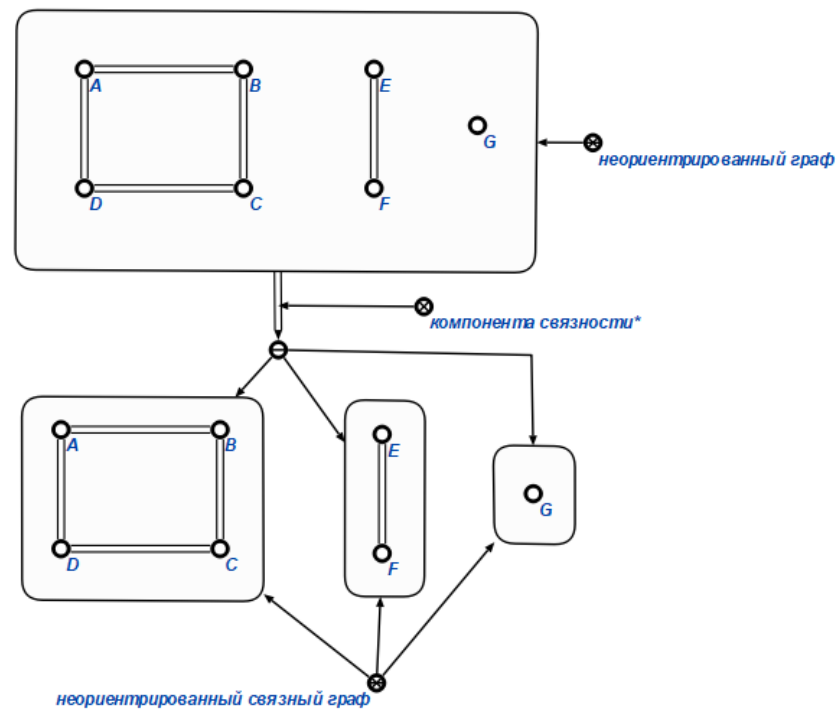


Рисунок 1.22 – Шаг 11

## **2 Личный вклад в развитие ИС по искусству**

### **2.1 Список формализованных понятий**

Был дополнен раздел "Театр". В рамках этого раздела были формализованы следующие понятия:

1. Относительные понятия
  - 1.1. Звукорежиссер'
  - 1.2. Суфлер'
  - 1.3. Реплика\*
2. Абсолютные понятия
  - 2.1. Ансамбль
  - 2.2. Сцена
  - 2.3. Реквизит
  - 2.4. Театральная публика
  - 2.5. Занавес
  - 2.6. Костюм
  - 2.7. Грим
  - 2.8. Гардероб
  - 2.9. Театральное действие (акт)
  - 2.10. Бутафория (бутафорский предмет)
  - 2.11. Текст роли/слова актера
  - 2.12. Зрительный зал
3. Вспомогательные понятия (которые были формализованы лишь частично и не обязательно относятся к предметной области)
  - 3.1. Руководитель'
  - 3.2. Обмен\*
  - 3.3. Место\*
  - 3.4. Подсказка\*
  - 3.5. Шептать\*
  - 3.6. Необходимость\*
  - 3.7. Ключевая часть\*
  - 3.8. Дешевизна\*
  - 3.9. Непрочность\*
  - 3.10. Книга
  - 3.11. Совокупность
  - 3.12. Музыкальный коллектив
  - 3.13. Предмет
  - 3.14. Подлинная вещь
  - 3.15. Помещение
  - 3.16. Текст
  - 3.17. Звук

- 3.18. Глава
- 3.19. Монолог
- 3.20. Диалог
- 3.21. Лицо
- 3.22. Партер
- 3.23. Амфитеатр
- 3.24. Бенуар
- 3.25. Бельэтаж
- 3.26. Балкон
- 3.27. Эмблема
- 3.28. Одежда

## 2.2 Примеры реализации формализованных понятий

### 1. Грим

Рассмотрим формализованное на языке SC абсолютное понятие "Грим". На рисунке 2.1 изображен фрагмент базы знаний ИС по искусству, показывающий идентификаторы понятия "Грим".

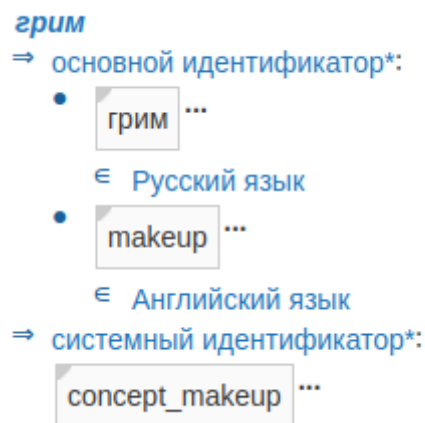


Рисунок 2.1 – Идентификаторы понятия "Грим"

На рисунке 2.2 изображен фрагмент базы знаний ИС по искусству, показывающий утверждение и определение понятия "Грим".

- Утв.(грим, режиссёр, спектакль, актер)

⇒ используемые константы:

...

⇐ трансляция sc-текста\*:

...

⇒ пример\*:

Характер грима в театре зависит от художественных особенностей пьесы, замысла актёра, режиссёрской концепции и стиля оформления спектакля.

∈ Русский язык

- Опр. (грим)

⇒ используемые константы:

...

⇐ трансляция sc-текста\*:

...

⇒ пример\*:

Грим - необходимое для игры на сцене оформление лица актёра.

∈ Русский язык

Рисунок 2.2 – Утверждение и определение понятия "Грим".

На рисунке 2.3 изображен фрагмент базы знаний ИС по искусству, показывающий рисунок понятия "Грим".

∈ ключевой sc-элемент\*:

- Рис. (Грим)

⇐ трансляция sc-текста\*:

...

⇒ пример\*:



Рисунок 2.3 – Рисунок понятия "Грим".

## 2. Бутафория

Рассмотрим формализованное на языке SC абсолютное понятие "Бутафория".

На рисунке 2.4 изображен фрагмент базы знаний ИС по искусству, показывающий идентификаторы понятия "Бутафория".

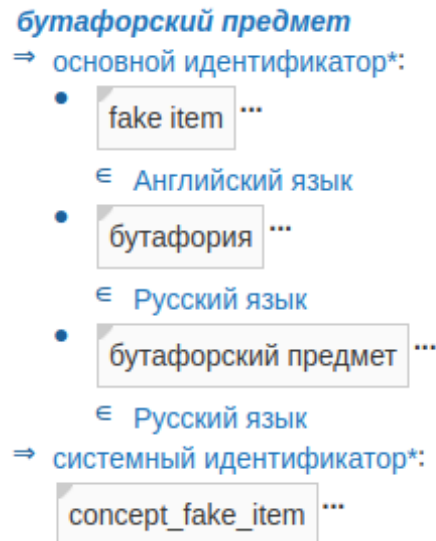


Рисунок 2.4 – Идентификаторы понятия "Бутафория"

На рисунке 2.5 изображен фрагмент базы знаний ИС по искусству, показывающий утверждение и определение понятия "Бутафория", а также принадлежность к множеству "Подлинность".

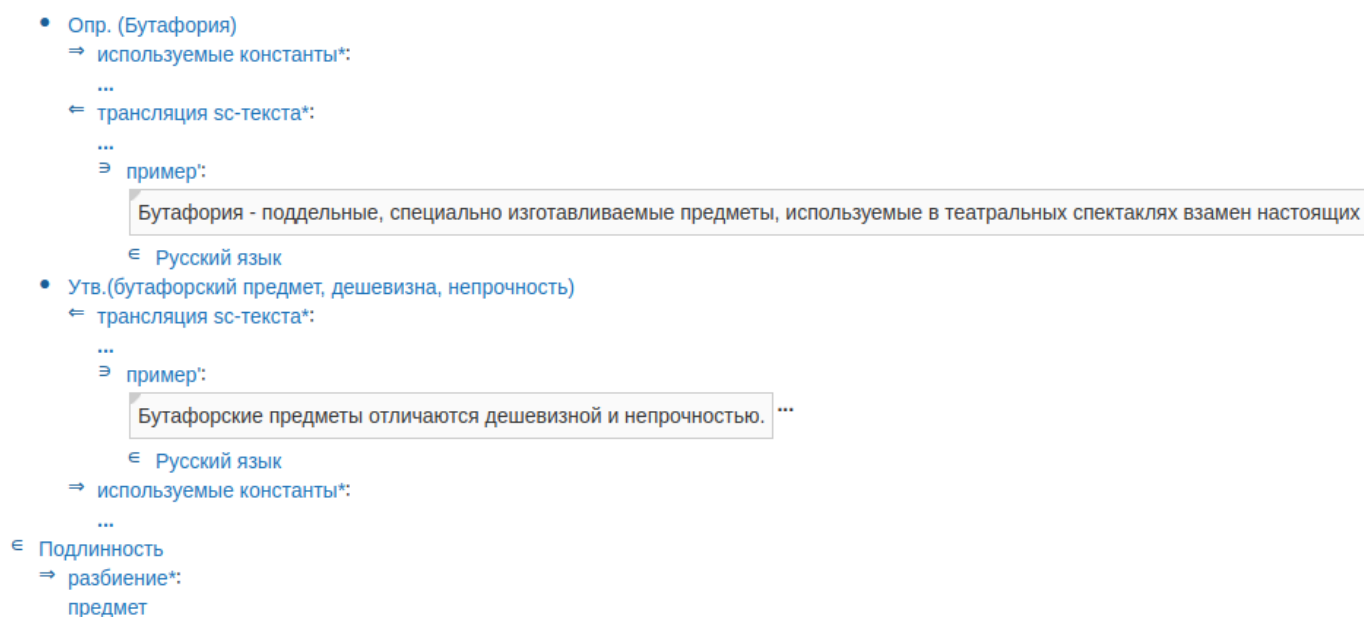


Рисунок 2.5 – Утверждение, определение и принадлежность к множеству "Подлинность" понятия "Бутафория".

На рисунке 2.6 изображен фрагмент базы знаний ИС по искусству, показывающий рисунок понятия "Бутафория".

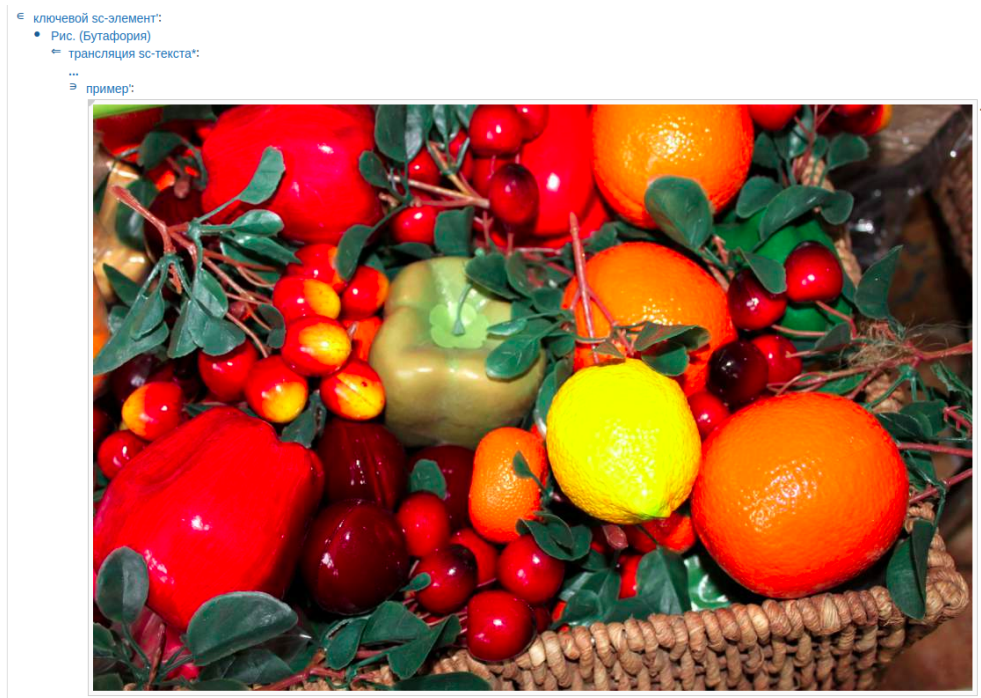


Рисунок 2.6 – Рисунок понятия "Бутафория".

### 3. Предмет

Рассмотрим формализованное на языке SC абсолютное понятие "Предмет". Оно не принадлежит предметной области "искусство но необходимо в качестве вспомогательного понятия для формализации понятий, принадлежащих предметной области (пример: Бутафория). На рисунке 2.7 изображен фрагмент базы знаний ИС по искусству, показывающий формализацию понятия "Предмет".

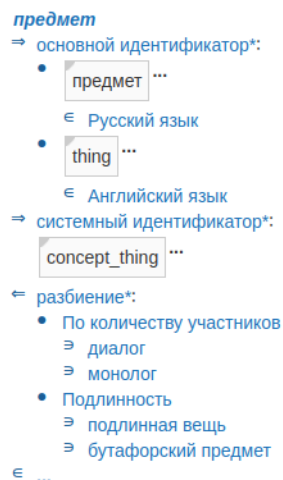


Рисунок 2.7 – Абсолютное понятие "Предмет" в ИС по искусству

## Заключение

В рамках курсовой работы были решены следующие задачи:

- Разработан алгоритм нахождения компонент связности неориентированного графа;
- Разработанный алгоритм реализован с использованием библиотеки, поддерживающей SC-память, на классическом ЯП (C++);
- Разработанный алгоритм реализован в виде программы на ЯП SCp: языке, ориентированном на обработку семантических сетей.
- Внесен личный вклад в развитие ИС по искусству. В рамках раздела "Театр" было формализовано 15 понятий (3 относительных и 12 абсолютных), а также 28 вспомогательных понятий.



## Список использованных источников

- [1] Метасистема IMS. <http://ims.ostis.net>.
- [2] В.А., Горбатов. Фундаментальные основы дискретной математики. Информационная математика / Горбатов В.А. — Наука, Физматлит, 2000. — 544 с.
- [3] Ф.А., Новиков. Дискретная математика для программистов / Новиков Ф.А. — Питер, 2003. — 364 с.
- [4] В.Б., Тарасов. От многоагентных систем к интеллектуальным организациям / Тарасов В.Б. — Изд-во УРСС, 2002.
- [5] О., Оре. Теория графов / Оре О. — Наука, 1980. — 336 с.
- [6] Головкин, В. А. Нейроинтеллект: теория и применение. Книга 1: Организация и обучение нейронных сетей с прямыми и обратными связями / В. А. Головкин. — Брестский политехнический институт., 1999. — 265 Р.
- [7] Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика / Борисов В.В. Круглов В.В. — Горячая линия-Телеком., 2002. — 383 Р.
- [8] Зыков, А.А. Основы теории графов / А.А. Зыков. — Издательство «Наука», 1987. — 384 с.
- [9] Харарри, Ф. Теория графов / Ф. Харарри. — Эдиториал УРСС, 2018. — 304 Р.
- [10] Оре, О. Теория графов / О. Оре. — Наука, 1980. — 336 Р.
- [11] Кузнецов, О. П. Дискретная математика для инженера / О. П. Кузнецов, Г. М. Адельсон-Вельский. — Энергоатомиздат, 1988. — 480 Р.
- [12] Кормен, Д. Алгоритмы. Построение и анализ / Д. Кормен. — Вильямс, 2015. — 1328 Р.
- [13] Нечепуренко М.И., Попков В.К. Алгоритмы и программы решения задач на графах и сетях / Попков В.К. Нечепуренко М.И. — Наука. Сиб. отд-ние, 1990. — 515 Р.
- [14] Емеличев В.А., Мельников О.И. Лекции по теории графов / Мельников О.И. Емеличев В.А. — Наука, 1990. — 384 Р.